CLAIM

We claim:

1. A packet monitor for examining packets passing through a connection point on a computer network in real-time, the packets provided to the packet monitor via a packet acquisition device connected to the connection point, the packet monitor comprising:

(a) a packet-buffer memory configured to accept a packet from the packet acquisition device;

(b) a parsing/extraction operations memory configured to store a database of parsing/extraction operations that includes information describing how to determine at least one of the protocols used in a packet from data in the packet;

(c) a parser subsystem coupled to the packet buffer and to the pattern/extraction operations memory, the parser subsystem configured to examine the packet accepted by the buffer, extract selected portions of the accepted packet, and form a function of the selected portions sufficient to identify that the accepted packet is part of a conversational flow-sequence;

(d) a memory storing a flow-entry database including a plurality of flow-entries for conversational flows encountered by the monitor;

(e) a lookup engine connected to the parser subsystem and to the flow-entry database, and configured to determine using at least some of the selected portions of the accepted packet if there is an entry in the flow-entry database for the conversational flow sequence of the accepted packet;

(f) a state patterns/operations memory configured to store a set of predefined state transition patterns and state operations such that traversing a particular transition pattern as a result of a particular conversational flow-sequence of packets indicates that the particular conversational flow-sequence is associated with the operation of a particular application program, visiting each state in a traversal including carrying out none or more predefined state operations;

(g) a protocol/state identification mechanism coupled to the state patterns/operations memory and to the lookup engine, the protocol/state identification engine configured to determine the protocol and state of the conversational flow of the packet; and

(h) a state processor coupled to the flow-entry database, the protocol/state identification engine, and to the state patterns/operations memory, the state processor, configured to carry out any state operations specified in the state patterns/operations memory for the protocol and state of the flow of the packet,

the carrying out of the state operations furthering the process of identifying which application program is associated with the conversational flow-sequence of the packet, the state processor progressing through a series of states and state operations until there are no more state operations to perform for the accepted packet, in which case the state processor updates the flow-entry, or until a final state is reached that indicates that no more analysis of the flow is required, in which case the result of the analysis is announced.

2. A packet monitor according to claim 1, wherein the flow-entry includes the state of the flow, such that the protocol/state identification mechanism determines the state of the packet from the flow-entry in the case that the lookup engine finds a flow-entry for the flow of the accepted packet.

3. A packet monitor according to claim 1, wherein the parser subsystem includes a mechanism for building a hash from the selected portions, and wherein the hash is used by the lookup engine to search the flow-entry database, the hash designed to spread the flow-entries across the flow-entry database.

4. A packet monitor according to claim 1, further comprising:

a compiler processor coupled to the parsing/extraction operations memory, the compiler processor configured to run a compilation process that includes:

receiving commands in a high-level protocol description language that describe the protocols that may be used in packets encountered by the monitor, and

translating the protocol description language commands into a plurality of parsing/extraction operations that are initialized into the parsing/extraction operations memory.

5. A packet monitor according to claim 4, wherein the protocol description language commands also describe a correspondence between a set of one or more application programs and the state transition patterns/operations that occur as a result of particular conversational flow-sequences associated with an application program, wherein the compiler processor is also coupled to the state patterns/operations memory, and wherein the compilation process further includes translating the protocol description language commands into a plurality of state patterns and state operations that are initialized into the state patterns/operations memory.

6. A packet monitor according to claim 1, further comprising:

a cache memory coupled to and between the lookup engine and the flow-entry database providing for fast access of a set of likely-to-be-accessed flow-entries from the flow-entry database.

7. A packet monitor according to claim 6, wherein the cache functions as a fully associative, least-recently-used cache memory.

8. A packet monitor according to claim 7, wherein the cache functions as a fully associative, least-recently-used cache memory and includes content addressable memories configured as a stack.

9. A packet monitor according to claim 1, wherein one or more statistical measures about a flow are stored in each flow-entry, the packet monitor further comprising:

a calculator for updating the statistical measures in a flow-entry of the accepted packet.

10. A packet monitor according to claim 9, wherein, when the application program of a flow is determined, one or more network usage metrics related to said application and determined from the statistical measures are presented to a user for network performance monitoring.

11. A method of examining packets passing through a connection point on a computer network, each packets conforming to one or more protocols, the method comprising:

(a) receiving a packet from a packet acquisition device;

(b) performing one or more parsing/extraction operations on the packet to create a parser record comprising a function of selected portions of the packet;

(c) looking up a flow-entry database comprising none or more flow-entries for previously encountered conversational flows, the looking up using at least some of the selected packet portions and determining if the packet is of an existing flow;

(d) if the packet is of an existing flow, classifying the packet as belonging to the found existing flow; and

(e) if the packet is of a new flow, storing a new flow-entry for the new flow in the flow-entry database, including identifying information for future packets to be identified with the new flow-entry,

wherein the parsing/extraction operations depend on one or more of the protocols to which the packet conforms.

12. A method according to claim 11, wherein each packet passing through the connection point is examined in real time.

13. A method according to claim 11, wherein classifying the packet as belonging to the found existing flow includes updating the flow-entry of the existing flow.

14. A method according to claim 13, wherein updating includes storing one or more statistical measures stored in the flow-entry of the existing flow.

15. A method according to claim 14, wherein the one or more statistical measures include measures selected from the set consisting of the total packet count for the flow, the time, and a differential time from the last entered time to the present time.

16. A method according to claim 11, wherein the function of the selected portions of the packet forms a signature that includes the selected packet portions and that can identify future packers, wherein the lookup operation uses the signature and wherein the

identifying information stored in the new or updated flow-entry is a signature for identifying future packets.

17. A method according to claim 11, wherein at least one of the protocols of the packet uses source and destination addresses, and wherein the selected portions of the packet include the source and destination addresses.

18. A method according to claim 17, wherein the function of the selected portions for packets of the same flow is consistent independent of the direction of the packets.

19. A method according to claim 18, wherein the source and destination addresses are placed in an order determined by the order of numerical values of the addresses in the function of selected portions.

20. A method according to claim 19, wherein the numerically lower address is placed before the numerically higher address in the function of selected portions.

21. A method according to claim 11, wherein the looking up of the flow-entry database uses a hash of the selected packet portions.

22. A method according to claim 11, wherein the parsing/extraction operations are according to a database of parsing/extraction operations that includes information describing how to determine a set of one or more protocol dependent extraction operations from data in the packet that indicate a protocol used in the packet.

23. A method according to claim 11, wherein step (d) includes if the packet is of an existing flow, obtaining the last encountered state of the flow and performing any state operations specified for the state of the flow starting from the last encountered state of the flow; and wherein step (e) includes if the packet is of a new flow, performing any state operations required for the initial state of the new flow.

24. A method according to claim 23, wherein the state processing of each received packet of a flow furthers the identifying of the application program of the flow.

25. A method according to claim 23, wherein the state operations include updating the flow-entry, including storing identifying information for future packets to be identified with the flow-entry.

26. A method according to claim 25, wherein the state processing of each received packet of a flow furthers the identifying of the application program of the flow.

27. A method according to claim 23, wherein the state operations include searching the parser record for the existence of one or more reference strings.

28. A method according to claim 23, wherein the state operations are carried out by a programmable state processor according to a database of protocol dependent state operations.

29. A packet monitor for examining packets passing through a connection point on a computer network, each packets conforming to one or more protocols, the monitor comprising:

(a) a packet acquisition device coupled to the connection point and configured to receive packets passing through the connection point;

(b) an input buffer memory coupled to and configured to accept a packet from the packet acquisition device;

(c) a parser subsystem coupled to the input buffer memory and including a slicer, the parsing subsystem configured to extract selected portions of the accepted packet and to output a parser record containing the selected portions;

(d) a memory for storing a database comprising none or more flow-entries for previously encountered conversational flows, each flow-entry identified by identifying information stored in the flow-entry;

(e) a lookup engine coupled to the output of the parser subsystem and to the flow-entry memory and configured to lookup whether the particular packet whose parser record is output by the parser subsystem has a matching flow-entry, the looking up using at

least some of the selected packet portions and determining if the packet is of an existing flow; and

(f) a flow insertion engine coupled to the flow-entry memory and to the lookup engine and configured to create a flow-entry in the flow-entry database, the flow-entry including identifying information for future packets to be identified with the new flow-entry,

the lookup engine configured such that if the packet is of an existing flow, the monitor classifies the packet as belonging to the found existing flow; and if the packet is of a new flow, the flow insertion engine stores a new flow-entry for the new flow in the flow-entry database, including identifying information for future packets to be identified with the new flow-entry,

wherein the operation of the parser subsystem depends on one or more of the protocols to which the packet conforms.

30. A monitor according to claim 29, wherein each packet passing through the connection point is accepted by the packet buffer memory and examined by the monitor in real time.

31. A monitor according to claim 29, wherein the lookup engine updates the flow-entry of an existing flow in the case that the lookup is successful.

32. A monitor according to claim 29, further including a mechanism for building a hash from the selected portions, wherein the hash is included in the input for a particular packet to the lookup engine, and wherein the hash is used by the lookup engine to search the flow-entry database.

33. A monitor according to claim 29, further including a memory containing a database of parsing/extraction operations, the parsing/extraction database memory coupled to the parser subsystem, wherein the parsing/extraction operations are according to one or more parsing/extraction operations looked up from the parsing/extraction database.

34. A monitor according to claim 33, wherein the database of parsing/extraction operations includes information describing how to determine a set of one or more

protocol dependent extraction operations from data in the packet that indicate a protocol used in the packet.

35. A monitor according to claim 29, further including a flow-key-buffer (UFKB) coupled to the output of the parser subsystem and to the lookup engine and to the flow insertion engine, wherein the output of the parser monitor is coupled to the lookup engine via the UFKB, and wherein the flow insertion engine is coupled to the lookup engine via the UFKB.

36. A method according to claim 29, further including a state processor coupled to the lookup engine and to the flow-entry-database memory, and configured to perform any state operations specified for the state of the flow starting from the last encountered state of the flow in the case that the packet is from an existing flow, and to perform any state operations required for the initial state of the new flow in the case that the packet is from an existing flow.

37. A method according to claim 29, wherein the set of possible state operations that the state processor is configured to perform includes searching for one or more patterns in the packet portions.

38. A monitor according to claim 36, wherein the state processor is programmable, the monitor further including a state patterns/operations memory coupled to the state processor, the state operations memory configured to store a database of protocol dependent state patterns/operations.

39. A monitor according to claim 35, further including a state processor coupled to the UFKB and to the flow-entry-database memory, and configured to perform any state operations specified for the state of the flow starting from the last encountered state of the flow in the case that the packet is from an existing flow, and to perform any state operations required for the initial state of the new flow in the case that the packet is from an existing flow.

40. A monitor according to claim 36, wherein the state operations include updating the flow-entry, including identifying information for future packets to be identified with the flow-entry.

41. A packet monitor according to claim 29, further comprising:

a compiler processor coupled to the parsing/extraction operations memory, the compiler processor configured to run a compilation process that includes:

receiving commands in a high-level protocol description language that describe the protocols that may be used in packets encountered by the monitor and any children protocols thereof, and

translating the protocol description language commands into a plurality of parsing/extraction operations that are initialized into the parsing/extraction operations memory.

42. A packet monitor according to claim 38, further comprising:

a compiler processor coupled to the parsing/extraction operations memory, the compiler processor configured to run a compilation process that includes:

receiving commands in a high-level protocol description language that describe a correspondence between a set of one or more application programs and the state transition patterns/operations that occur as a result of particular conversational flow-sequences associated with an application programs, and

translating the protocol description language commands into a plurality of state patterns and state operations that are initialized into the state patterns/operations memory.

43. A packet monitor according to claim 29, further comprising:

a cache subsystem coupled to and between the lookup engine and the flow-entry database memory providing for fast access of a set of likely-to-be-accessed flow-entries from the flow-entry database.

44. A packet monitor according to claim 43, wherein the cache subsystem is an associative cache subsystem including one or more content addressable memory cells (CAMs).

45. A packet monitor according to claim 44, wherein the cache subsystem is also a least-recently-used cache memory such that a cache miss updates the least recently used cache entry.

46. A packet monitor according to claim 29, wherein each flow-entry stores one or more statistical measures about the flow, the monitor further comprising

a calculator for updating at least one of the statistical measures in the flow-entry of the accepted packet.

47. A packet monitor according to claim 46, wherein the one or more statistical measures include measures selected from the set consisting of the total packet count for the flow, the time, and a differential time from the last entered time to the present time.

48. A packet monitor according to claim 46, further including a statistical processor configured to determine one or more network usage metrics related to the flow from one or more of the statistical measures in a flow-entry.

49. A monitor according to claim 29, wherein:

flow-entry-database is organized into a plurality of bins that each contain N-number of flow-entries, and wherein said bins are accessed via a hash data value created by a parser subsystem based on the selected packet portions, wherein N is one or more.

50. A monitor according to claim 49, wherein the hash data value is used to spread a plurality of flow-entries across the flow-entry-database and allows fast lookup of a flow-entry and shallower buckets.

51. A monitor according to claim 36, wherein the state processor analyzes both new and existing flows in order to classify them by application and proceeds from state-to-state based on a set of predefined rules.

52. A monitor according to claim 29, wherein the lookup engine begins processing as soon as a parser record arrives from the parser subsystem.

53. A monitor according to claim 36, wherein the lookup engine provides for flow state entry checking to see if a flow key should be sent to the state processor, and that outputs a protocol identifier for the flow.

54. A method of examining packets passing through a connection point on a computer network, the method comprising:

(a) receiving a packet from a packet acquisition device;

(b) performing one or more parsing/extraction operations on the packet according to a database of parsing/extraction operations to create a parser record comprising a function of selected portions of the packet, the database of parsing/extraction operations including information on how to determine a set of one or more protocol dependent extraction operations from data in the packet that indicate a protocol is used in the packet;

(c) looking up a flow-entry database comprising none or more flow-entries for previously encountered conversational flows, the looking up using at least some of the selected packet portions, and determining if the packet is of an existing flow;

(d) if the packet is of an existing flow, obtaining the last encountered state of the flow and performing any state operations specified for the state of the flow starting from the last encountered state of the flow; and

(e) if the packet is of a new flow, performing any analysis required for the initial state of the new flow and storing a new flow-entry for the new flow in the flow-entry database, including identifying information for future packets to be identified with the new flow-entry.

55. A method according to claim 54, wherein one of the state operations specified for at least one of the states includes updating the flow-entry, including identifying information for future packets to be identified with the flow-entry.

56. A method according to claim 54, wherein one of the state operations specified for at least one of the states includes searching the contents of the packet for at least one reference string.

57. A method according to claim 55, wherein one of the state operations specified for at least one of the states includes creating a new flow-entry for future packets to be identified with the flow, the new flow-entry including identifying information for future packets to be identified with the flow-entry.

58. A method according to claim 54, further comprising forming a signature from the selected packet portions, wherein the lookup operation uses the signature and wherein the identifying information stored in the new or updated flow-entry is a signature for identifying future packets.

59. A method according to claim 54, wherein the state operations are according to a database of protocol dependent state operations.